Visualization and Analysis for Near-Real-Time Decision Making in Distributed Workflows

David Pugmire
Oak Ridge National Laboratory

James Kress
University of Oregon &
Oak Ridge National Laboratory

Hank Childs
University of Oregon &
Lawrence Berkeley National Laboratory

Mathew Wolf, Greg Eisenhauer
Georgia Institute of Technology

Johnathan Low
A*STAR Computational Resource Centre

Randy Michael Churchill
Princeton Plasma Physics Laboratory

Tahsin Kurc
Stony Brook University &
Oak Ridge National Laboratory

Kesheng Wu, Alexander Sim, Junmin Gu
Lawrence Berkeley National Laboratory

Jong Choi, Scott Klasky
Oak Ridge National Laboratory

***Abstract*—Data driven science is becoming increasingly more common, complex, and is placing tremendous stresses on visualization and analysis frameworks. Data sources producing 10GB per second and more, are becoming increasingly commonplace in both simulation, sensor and experimental sciences. These data sources, which are typically distributed around the world, must be analyzed by teams of scientists that are also distributed. Enabling scientists to view, query and interact with such large volumes of data in near-real-time requires a rich fusion of visualization and analysis techniques, middleware and workflow systems. This paper discusses initial research into visualization and analysis of distributed data workflows that enables scientists to make near-real-time decisions of large volumes of time varying data.**

I. Introduction

Data driven sciences are placing enormous stresses on existing visualization and analysis frameworks. These stresses are occurring across several different axes.

First, increasingly larger volumes of data are being generated, and at increasing frequency as well. For example, in the test case that we are exploring, large scale parallel plasma fusion simulation codes, 10+ GB of data are being generated each second. Similar amounts and frequency of data are generated from sensors inside fusion experiments that are in operation already, and will only increase when large experiments, such as ITER [1] become fully operational.

Second, scientists and data are often distributed in multiple geographic locations. This requires that either the scientists be moved closer to the data, which is fraught with logistical and convenience difficulties, or move data (either all, or selected portions) to where the scientists are located.

The movement of data leads to additional stresses. It is often not possible to move the large volumes of data to remote locations such that scientists can interact, analyze and visualize it in a near-real-time manner that will enable them to make timely decisions. Finally, the workflows (both simple and complex) required by scientists can require computational resources that may not be available at the remote sites where the scientists are located.

In this paper, we describe initial work from an active research effort to explore data coupling and near-real-time analysis and visualization between two geographically separated sites. In this instance, Singapore and Atlanta. Specifically, a plasma fusion simulation running at the A*STAR Computational Resource Centre in Singapore, and the visualization and analysis running at Georgia Tech in Atlanta, Georgia. We demonstrated this working system at the SuperComputing 2015 Conference in Austin, Texas. At each timestep of the simulation running in Singapore, summary data would be generated and moved to Atlanta where it would be displayed by visualization tools. The scientist is then able to select regions of interest from the summary data and extract features from this area of interest. Once the features are identified, simulation particle data that lie within the features are extracted and moved to Atlanta for visualization.

In this work we are exploring fusion of a variety of different technologies. We are using a high-level API to provide location independent data access and remote reading and writing. The middleware components implement RDMA over wide-area networks and support data indexing for optimized filtering operations. The data analysis and visualization components use this middleware to facilitate rich interactions with the data. These components use the data subsetting and filtering operations of the middleware to achieve near-real-time interaction with the running simulation. These results show that near-real-time interaction can be achieved, even with the sites are separated by 10's of thousands of miles. Further, we are able to show end-to-end data selection and visualization within the tight 10 second time constraint window imposed by the running simulation.

In the remainder of this paper we discuss related work, discuss some of the broader motivations for this work, provide a detailed discussion of the system and the results obtained to date. Finally we discuss areas of continuing and future research.

II. Related Work

Past work in this area has explored both simulation monitoring and steering. A lot of past effort has gone into designing methods for quickly and efficiently visualizing data across a network. Some notable examples include Visapult [2], Visualization Dot Com [3], VisPortal [4], and a Real-Time Monitoring framework for large scientific simulations [5]. VisPortal and Visualization Dot Com build on the foundations of Visapult, and provide a remote distributed visualization framework for efficient visualization of remote simulation data. This framework uses both the local visualization client and the remote data client to perform parallel renderings, decreasing the time to produce the final visualizations. By leveraging Visapult, VisPortal and Visualization Dot Com are able to provide convenient access to simulation data to scientists through an easy to use and access online interface.

One notable example of work in simulation steering is SCIRun [6]. SCIRun presents a programming environment to simulation scientists and easily allows them to modify their simulations interactively as well as create automatically changing parameters based on boundary conditions.

Our work leverages the ideas from these past projects, and has allowed us to create a visualization and analysis pipeline that is extensible and operates on user driven subselections of live simulation data.

III. Objectives

Our objectives in this research are to explore data coupling and near-real-time analysis and visualization between timevarying producers of large data, and distributed data consumers. This capability for near-real-time access to data will help scientists observe, monitor, analyze the science as it happens, and enable them to make time-critical decisions.

We are working with the XGC1 [7] simulation code, a highly scalable physics code used to study plasmas in fusion tokamak devices. XGC1 is a particle-in-cell code, a common, and important method for solving physics problems. As such, XGC1 represents a large class of many different simulation codes. XGC1, like other particle-in-cell codes uses a grid, or mesh to represent a set of “cells”, and a large number of charged particles. At each timestep, the particles state is updated according to the underlying physics equations, and then the particles are statistically deposited onto the cells within the grid. Scientists are interested in both the mesh, and the particles.

In particular, the scientists are interested in the analysis and visualization of nonlinear turbulent eddies from XGC1, especially their 3D structure, and the perturbation to the orbits of particles within the eddies.

Because of the large volumes of data generated by XGC1, it serves as an excellent test-case for our research. XGC1 simulations routinely generate several TB’s of data, and larger runs, such as recent runs for ITER have produced 20 TB of data. Similarly, sensor networks attached to an experimental device like ITER are expected to generate TBs of data. Simulations and experiments in other domains produce similar amounts of data. Data volume estimates for the Square Kilometer Array Radio Telescope are even larger, around a TB of data every second.

Simulation centers and experimental facilities are scarce, and very expensive resources, and scientists have only fixed windows of time to do their science. Simulations that go awry, or encounter run-time problems translate into real loss of time to do science and the costs associated with running the facility. Experimental facilities face an even bigger problem. For instance, in the case of the ITER reactor, the buildup of instabilities within the plasma could cause physical damage to the reactor vessel. This results in significants costs for repairs, and downtime where other experiments are not able to run.

Allowing the scientists to remotely monitor and track their simulations and experiments in near-real-time will allow them to make important decisions. These include aborting when things appear to be going wrong, or not answering the anticipated questions being posed. Or to continue as the simulation or experiment is running as expected. And finally, to steer the simulation or experiment as the results for each timstep are observed and analyzed.

IV. System Implementation

Our visualization system is spread across two different geographic locations, The A*STAR Computational Resource Centre in Singapore and Georgia Tech in Atlanta, and is a synthesis of a variety of different tools and frameworks. On a high level this system consists of the simulation and data manipulation routines located in Singapore, the interactive visualization and analysis routines located at Georgia Tech, and a connection over the wide area network using the ADIOS [8] middleware and the the ICEE [9], FlexPath [10], and DataSpaces [11] transport methods. Combined, these technologies allowed us to perform memory to memory data delivery from one side of the pipeline to the other. The ADIOS middleware makes this delivery transparent to the simulation and visualization at either ends of the pipeline. A depiction of the system and data flow pipeline is shown in Fig. 1.

On the Georgia Tech side, our pipeline consists of two components, VisIt [12] to allow for interactive visualization and an eddy picker to allow the scientist to specify areas of interest. Python is used to coordinate communication between the VisIt and eddy picking tool, and perform some basic data analysis. The data picker was written in Qt and allows the user to select a region of interest on one of the planes of the simulation. In our case there were 512 individual planes spaced around the tokamak simulation, and are composites of the dpot (derivative potential) variable in the simulation, which enables the user to select areas of high turbulence for further visualization and analysis, see Fig. 2.

Eddies in turbulent fusion plasmas are typically elliptical in shape, so the picking tool supports selecting an ellipse with 3 points: the center point, and two points that lie on the ellipse. From these three points the major and minor radii are computed, as well as the direction of the major axis. Additionally, a dial is provided to control how many revolutions around the torus are used for construction of the 3D eddies.

Once a region of interest is selected, three things are triggered. First, the magnetic field line at the center point of

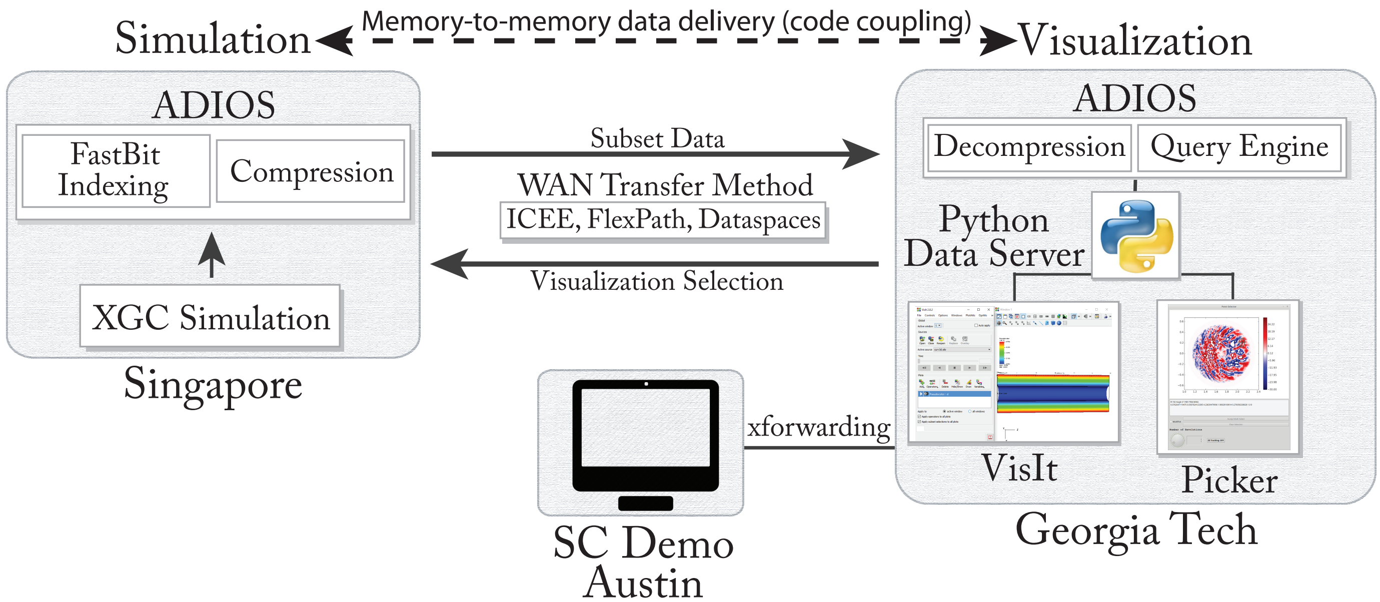

Fig. 1: The data flow pipeline for our workflow showing the distribution of the simulation and data querying in relation to the interactive visualization system.

the ellipse is calculated and displayed. Second, the 3D eddy feature is computed and displayed. These two steps are shown in Fig. 3. In this figure, a single plane from the simulation is shown, as well as the magnetic field line at the selection point (in yellow), and the 3D eddy (in orange).

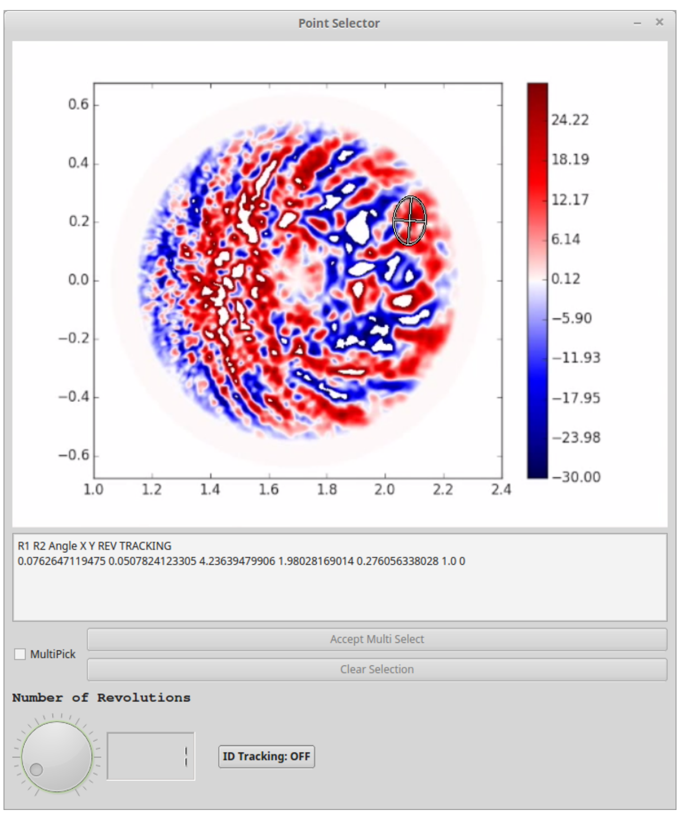

Fig. 2: The eddy selection interface demonstrating the selection of an eddy on a plane of the simulation data.

Next, a tight bounding volume of the 3D eddy is calculated and sent using ADIOS to Singapore. Once the Singapore side receives, using ADIOS, the bounding volume update, it will extract the particles that lie within the bounding volume. The extracted particles are sent using ADIOS to the Georgia Tech side and visualized. The particles, along with the magnetic field line and 3D eddy are shown visualized together in Fig. 4.

The scientists expressed the need to track particles contained in an eddy at a particular time and observe their evolution over time. This capability is activated with a toggle on the picker tool. If "ID Tracking" is turned on, then the IDs of the particles contained in the 3D eddy are cached, and at every timestep these particles are sent from Singapore to Georgia Tech for visualization. With this option, the scientists can follow the evolution of the particle orbits over time, and

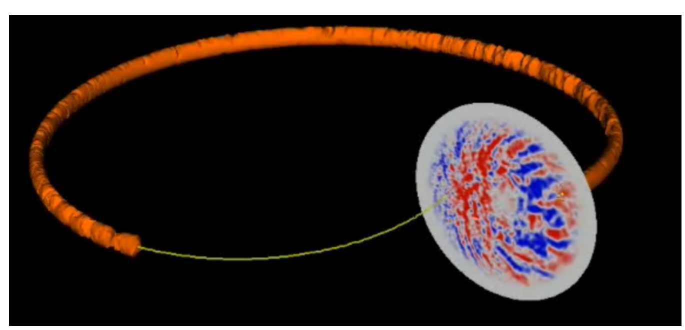

Fig. 3: The VisIt interface window demonstrating the tracking of the magnetic field line and eddy corresponding to the eddy selection in in the picker shown in 2.

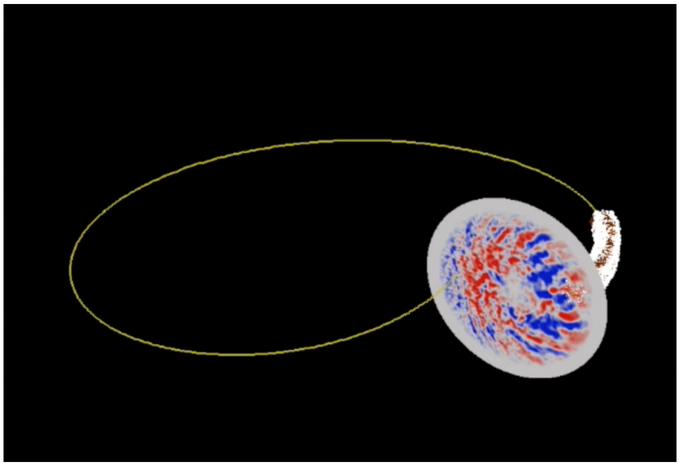

Fig. 4: The VisIt interface window demonstrating particle tracking and disbursement in a turbulent eddy over multiple simulation timesteps.

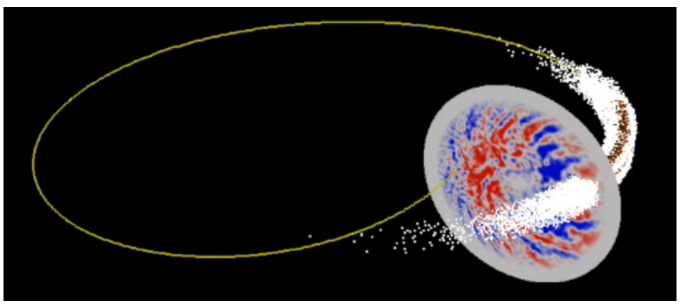

Fig. 5: The VisIt interface window demonstrating particle tracking by ID of particles that were inside a 3D eddy at a particular timestep in the past.

study their relationship to the field line and the 3D eddy feature. Fig. 5 shows an example of tracking particles by ID.

On the Singapore side, the pipeline has three primary components, the running XGC simulation, ADIOS, and the wide are network transfer method. The XGC1 simulation produces new timesteps at an average rate of once every 10 seconds. This data is then handled by the ADIOS middleware, which consists of two different paths of execution. First, the field variables used to calculate eddies are transmitted over the WAN to the visualization system so that it can begin processing the new timestep. Second, FastBit [13] indexing is done on the particle data from the new timestep. This indexing is done automatically for each new timestep so that if the particle data is requested by the visualization, it is indexed and ready to be transmitted based on the visualization query.

V. System Results and Performance

The performance and viability of our system was demonstrated at our booth on the SC15 Demo floor. We demonstrated that our system enabled near-real-time interaction with large data sets located around the world. Tests of our system were conducted between Singapore and Georgia Tech with Xwindow forwarding to the showroom floor. We gave our visualization and analysis routines a maximum of 10 seconds to perform an update. This time budget included the time to send 512 planes from Singapore to Georgia Tech, calculate the bounding boxes for the feature of interest on each, send that data back to Singapore, perform the data and particle subselection, send that data to Georgia Tech, and perform the visualization update. This maximum time limit kept us below the average time for a new timestep to be produced by the XGC simulation, allowing us to visualize each one as it was produced. Table I presents the size of the data being produced by the simulation, as well as the average data being sent to Georgia Tech after the user made a selection.

TABLE I: Breakdown of the data produced by XGC and processed by our pipeline during the course of the simulation.

Number of Planes	Number of Particles	Number of Time Steps
512	819,200,00	500
Particle Data Size	**3D dpot Data Size**	**Average Data per Selection**
62 GB per step	162 MB per step	0.1% subselection: 62 MB

The amount of data that we ended up having to send from Singapore back to Georgia Tech and process in our visualization routines is one of the main strengths of our system. By identifying the critical subsets of data, as defined by the scientist, we are able provide a near-real-time interactive experience with the running simulation. On average, the amount of particle data moved on each time step was around 62 MB, a mere 0.1% of the total data size.

Additonaly, this selection could be done very quickly though our use of FastBit to perform indexing on the simulation side. By having these indices precomputed, subselecting the data in Singapore was reduced to only a few seconds. This allowed our system to remain responsive to user update requests, and enabled new timesteps to be displayed as they were produced. This serves as a demonstration of a significant step forward in accomplishing our goal of a data driven, near-real-time, distributed visualization for a running simulation.

VI. Future Work

As this is an area of active research, we are planning on extending this work in a variety of different directions. First, we plan on using more complex workflows that utilize data from more sources. The work with fusion simulations can be extended to include experimental data, or previously run simulations for comparison. We will rely heavily on the ADIOS middleware to manage the complex, and time critical coordination and movement of data. Additionally, we will continue to work with the various transport methods in ADIOS to optimize performance. These more complex workflows, with different data sources, can employ machine learning techniques to detect features and events automatically. These methods will also serve as mechanisms for steering of simulations and experiments.

We also plan to explore workflows where in situ analysis and visualization are used as end products, or as pre-processing steps for other user defined tasks. As such, we plan to incorporate our previous work with EAVL and ADIOS [14] into these workflows.

For this particular demonstration of nonlinear turbulent eddies, we plan to use more advanced techniques for feature identification and extraction. This includes better identification of 2D features on each plane of the simulation, as well as the 3D construction of the eddies across a set of 2D planes. There is a wealth of research and development that can be utilized for better feature detection. Improved feature detection will allow for better identification of particles within the eddies, and aid in the study of their dynamical behavior in the plasma.

Finally, the infrastructures are largely science agnostic, and so working with additional simulations and experiments will provide opportunities for further expansion.

References

[1] European Fusion Development Agreement. The iter project, 2006. http://web.archive.org/web/20060824071609/http://www.efda.org/the_iter_project/index.htm.

[2] Wes Bethel. Visapult: A prototype remote and distributed visualization application and framework. *Lawrence Berkeley National Laboratory*, 2000.

[3] Wes Bethel. Visualization dot com. *Computer Graphics and Applications, IEEE*, 20(3):17–20, 2000.

[4] Wes Bethel, Cristina Siegerist, John Shalf, Praveenkumar Shetty, TJ Jankun-Kelly, Oliver Kreylos, and Kwan-Liu Ma. Visportal: Deploying grid-enabled visualization tools through a web-portal interface. *Lawrence Berkeley National Laboratory*, 2003.

[5] Valerio Pascucci, Daniel E Laney, Ray J Frank, Giorgio Scorzelli, Lars Linsen, Bernd Hamann, and Francois Gygi. Real-time monitoring of large scientific simulations. In *Proceedings of the 2003 ACM symposium on Applied computing*, pages 194–198. ACM, 2003.

[6] Steven G Parker and Christopher R Johnson. Scirun: a scientific programming environment for computational steering. In *Proceedings of the 1995 ACM/IEEE conference on Supercomputing*, page 52. ACM, 1995.

[7] CS Chang, S Ku, PH Diamond, Z Lin, S Parker, TS Hahm, and N Samatova. Compressed ion temperature gradient turbulence in diverted tokamak edgea). *Physics of Plasmas (1994-present)*, 16(5):056108, 2009.

[8] Qing Liu, Jeremy Logan, Yuan Tian, Hasan Abbasi, Norbert Podhorszki, Jong Youl Choi, Scott Klasky, Roselyne Tchoua, Jay Lofstead, Ron Oldfield, Manish Parashar, Nagiza Samatova, Karsten Schwan, Arie Shoshani, Matthew Wolf, Kesheng Wu, and Weikuan Yu. Hello adios: the challenges and lessons of developing leadership class i/o frameworks. *Concurrency and Computation: Practice and Experience*, 26(7):1453–1473, 2014.

[9] Jong Y Choi, Kesheng Wu, Jacky C Wu, Alex Sim, Qing G Liu, Matthew Wolf, C Chang, and Scott Klasky. Icee: Wide-area in transit data processing framework for near real-time scientific applications. In *4th SC Workshop on Petascale (Big) Data Analytics: Challenges and Opportunities in conjunction with SC13*, 2013.

[10] Jai Dayal, Drew Bratcher, Greg Eisenhauer, Karsten Schwan, Matthew Wolf, Xuechen Zhang, Hasan Abbasi, Scott Klasky, and Norbert Podhorszki. Flexpath: Type-based publish/subscribe system for large-scale science analytics. In *14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing(CCGRID '14)*, 2014.

[11] Ciprian Docan, Manish Parashar, and Scott Klasky. Dataspaces: an interaction and coordination framework for coupled simulation workflows. *Cluster Computing*, 15(2):163–181, 2012.

[12] Hank Childs, Eric Brugger, Brad Whitlock, Jeremy Meredith, Sean Ahern, David Pugmire, Kathleen Biagas, Mark Miller, Cyrus Harrison, Gunther H. Weber, Hari Krishnan, Thomas Fogal, Allen Sanderson, Christoph Garth, E. Wes Bethel, David Camp, Oliver Rübel, Marc Durant, Jean M. Favre, and Paul Navrátil. VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data. In *High Performance Visualization–Enabling Extreme-Scale Scientific Insight*, pages 357–372. Oct 2012.

[13] K. Wu, S. Ahern, E. W. Bethel, J. Chen, H. Childs, E. Cormier-Michel, C. G. R. Geddes, J. Gu, H. Hagen, B. Hamann, W. Koegler, J. Laurent, J. Meredith, P. Messmer, E. Otoo, V. Perevoztchikov, A. Poskanzer, Prabhat, O. Rübel, A. Shoshani, A. Sim, K. Stockinger, G. Weber, and W.-M. Zhang. FastBit: Interactively Searching Massive Data. In *Journal of Physics Conference Series, Proceedings of SciDAC*, volume 180, page 012053, San Diego, CA, June 2009.

[14] David Pugmire, James Kress, Jeremy Meredith, Norbert Podhorszki, Jong Choi, and Scott Klasky. Towards scalable visualization plugins for data staging workflows. In *Big Data Analytics: Challenges and Opportunities (BDAC-14) Workshop at Supercomputing Conference*, November 2014.